#_ Core Shell

Nikhil Jha

(Based on slides from Daniel Kessler, Douglas Wong, Sahil Hasan, Stefan Kuklinsky)

Logistics

- Lab 3 Released @ decal.ocf.io
- Ask questions in #decal-general on the OCF chat network.
  - ocf.io/slack
  - ocf.io/discord
  - ocf.io/irc
  - ocf.io/matrix

Today

- SSH
- Files, Pipes, and Redirection
- Tmux
- Vim

Shell

Why use the shell?

- Can make you more productive
- Easier to automate - do work without lifting a finger!
- Sometimes necessary

Is it unethical for me to not tell my employer I've automated my job?

741

248

I currently work on a legacy system for a company. The system is really old - and although I was hired as a programmer, my job is pretty much glorified data entry. To summarise, I get a bunch of requirements, which is literally just lots of data for each month on spreadsheets and I have to configure the system to make it work, which is basically just writing a whole bunch of SQL scripts.

It's not quite as simple as that, because whoever wrote the system originally really wrote it backwards, and in fact, the analysts who create the spreadsheets actually spend a fair bit of time verifying my work because the process is so tedious that it's easy to make a mistake.

As you can guess, it is pretty much the most boring job ever. However, it's a full time job with decent pay, and I work remotely so I can stay home with my son.

So I've been doing it for about 18 months and in that time, I've basically figured out all the traps to the point where I've actually written a program which for the past 6 months has been just doing the whole thing for me. So what used to take the last guy like a month, now takes maybe 10 minutes to clean the spreadsheet and run it through the program.

https://workplace.stackexchange.com/q/93696

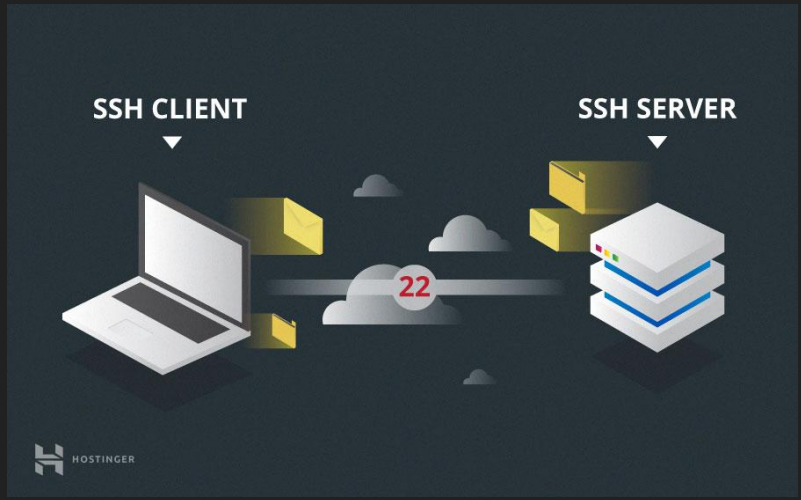
Common Shells

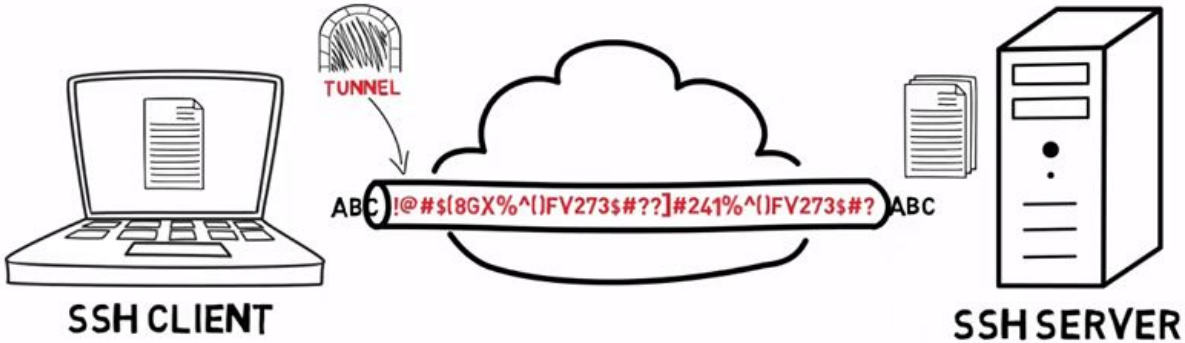
- Bash (Bourne-again shell)
- Zsh (Z shell)

- Bunch of other less common ones (dash, csh, tcsh, **fish**, etc.)

SSH (Secure Shell)

- Lets you log in to a remote computer
- `ssh <user>@<remote>`




TUNNEL
ABC
!@#$(8GX%^()FV273$#??]#241%^()FV273$#?
ABC
SSH CLIENT
SSH SERVER

Basic File management

- cp [source] [target]
- mv [source] [target]

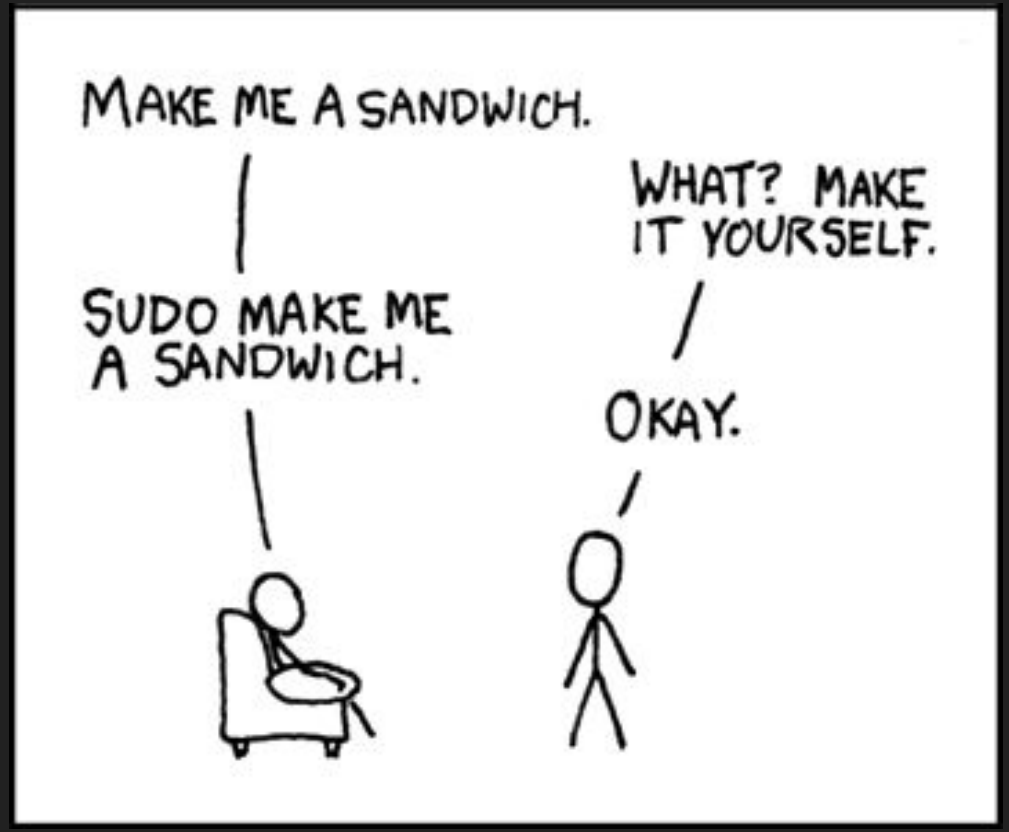
man [command]

Moving around the filesystem

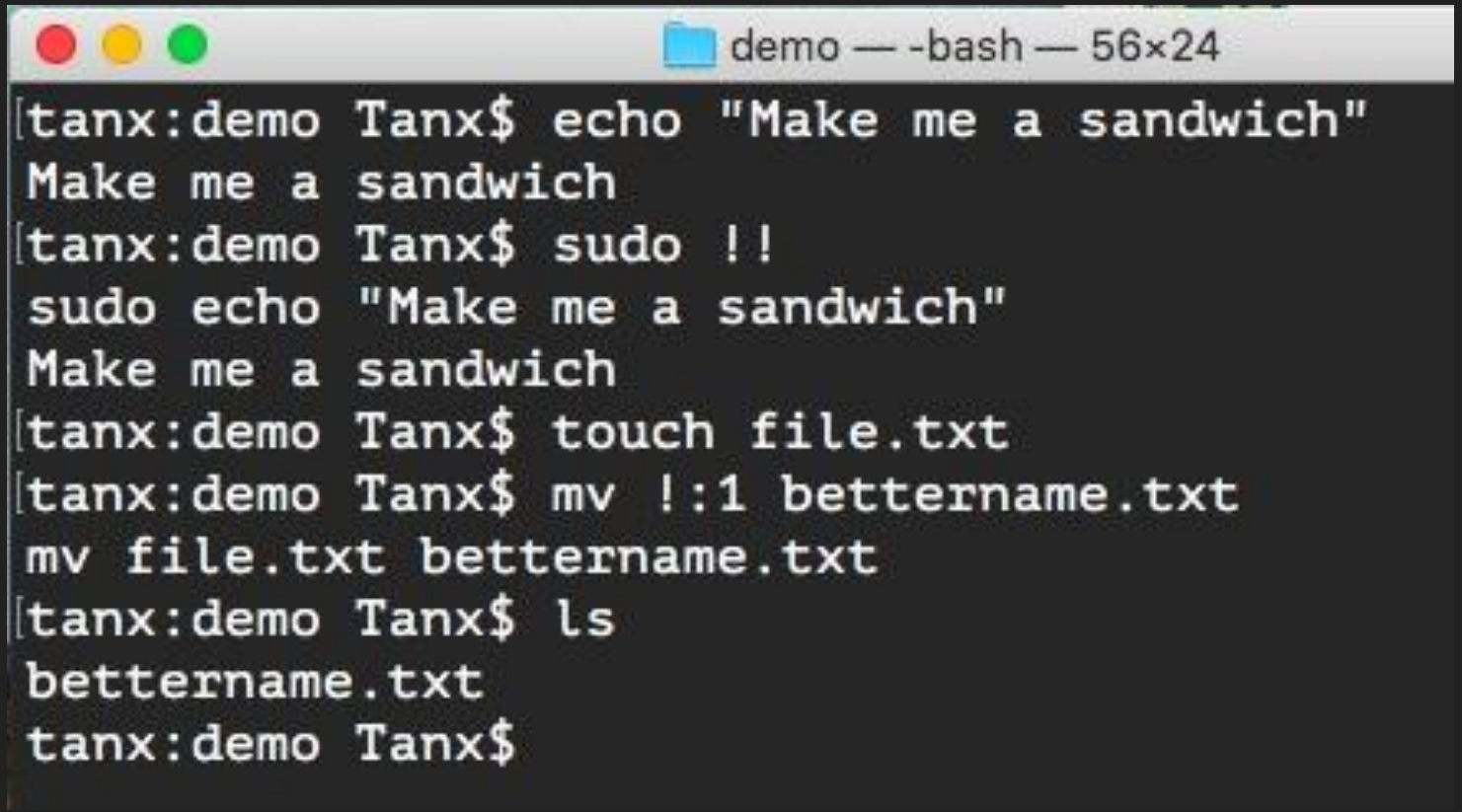
- Two ways of referencing a file: absolute and relative
  - /home/t/ta/tanx/file
  - ~/file
  - .
  - ..
- Absolute paths always start with /

Efficient shell

- Replaying commands: !!
- Argument substitution: !:1
  - Grab last argument: !$


MAKE ME A SANDWICH.
SUDO MAKE ME A SANDWICH.
WHAT? MAKE IT YOURSELF.
OKAY.

Demo


demo — -bash — 56×24
tanx:demo Tanx$ echo "Make me a sandwich"
Make me a sandwich
tanx:demo Tanx$ sudo !!
sudo echo "Make me a sandwich"
Make me a sandwich
tanx:demo Tanx$ touch file.txt
tanx:demo Tanx$ mv !:1 bettername.txt
mv file.txt bettername.txt
tanx:demo Tanx$ ls
bettername.txt
tanx:demo Tanx$

Pipe

- | (called a pipe)
- Utility for chaining commands together

cat people_names

cat people_names | grep Anna

Redirect

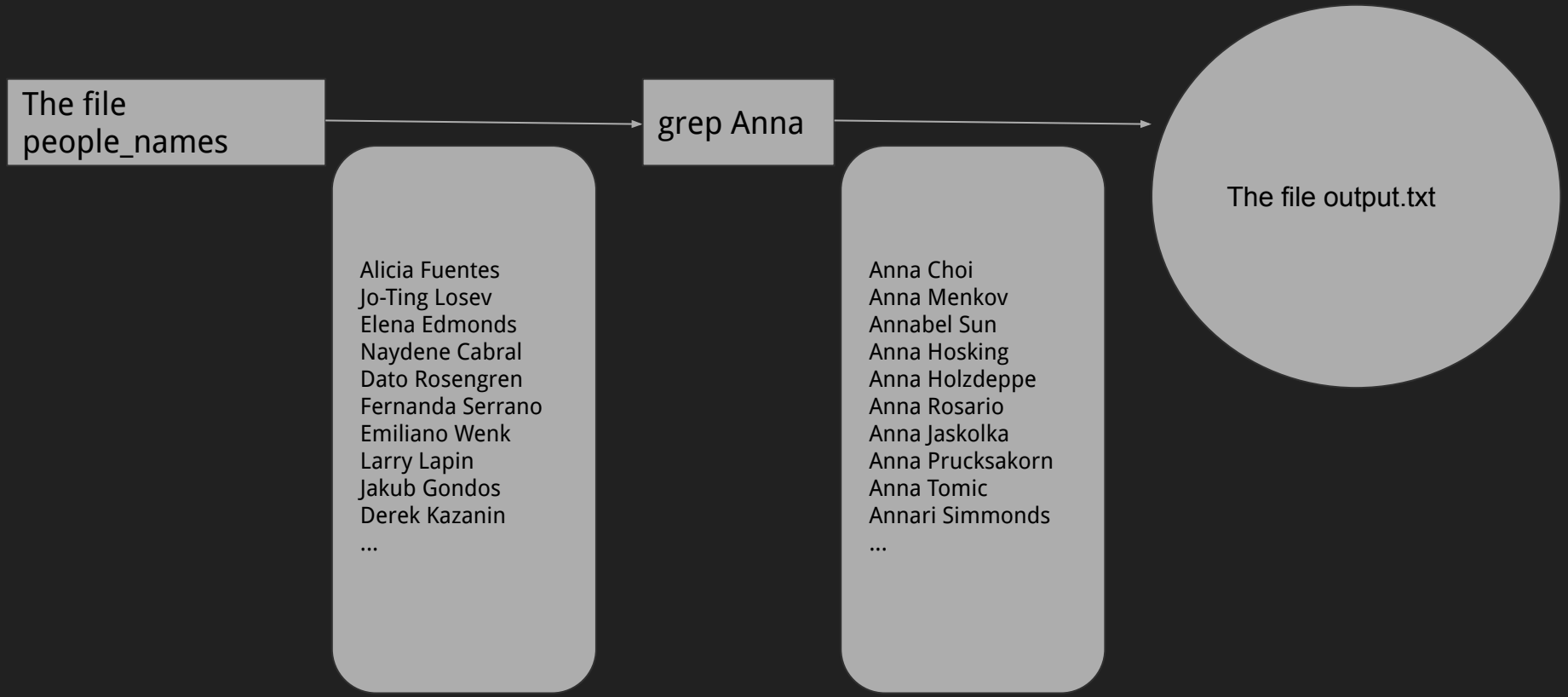
- stdin, stdout
- >
- <

cat people_names > test.txt

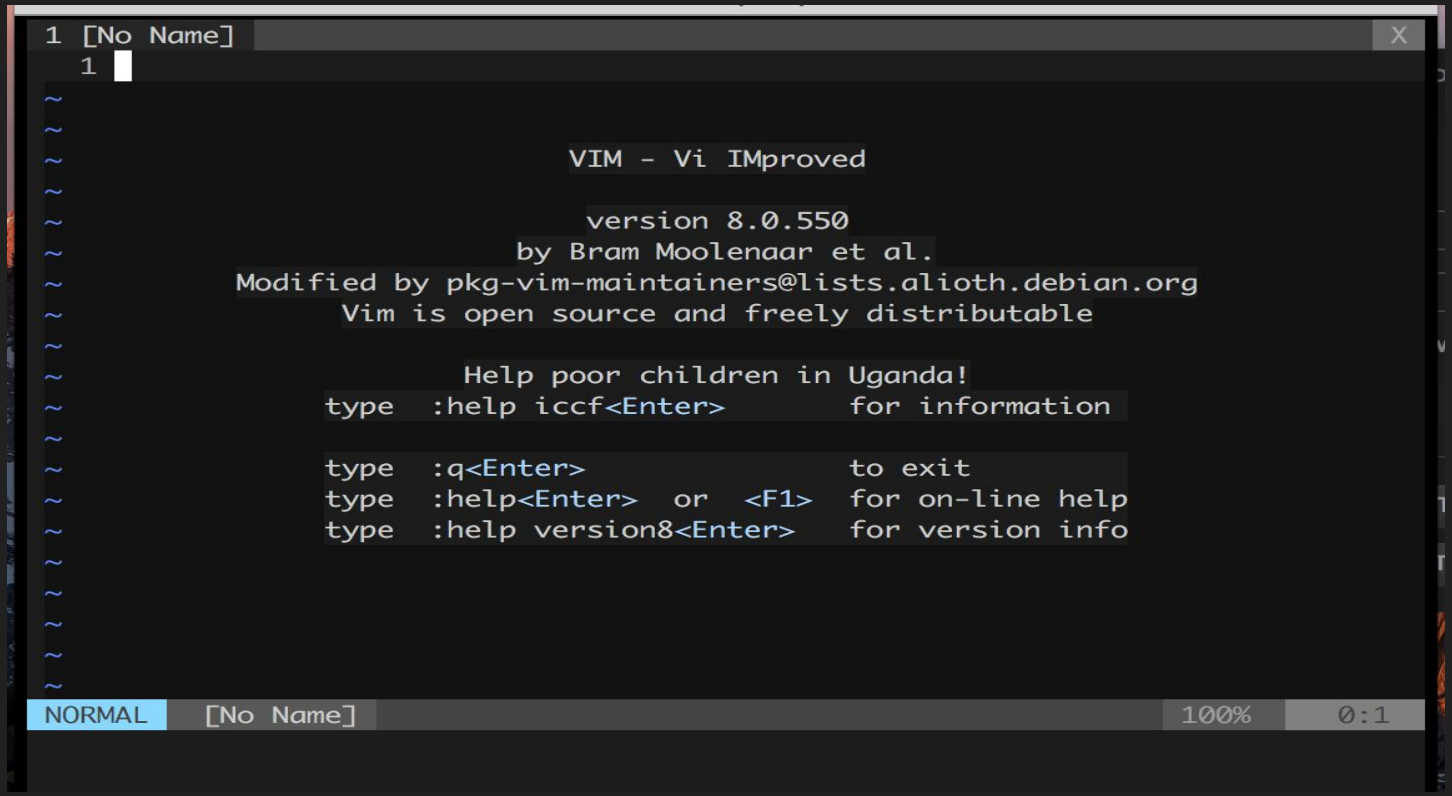
cat people_names

The file test.txt

grep Anna < people_names > output.txt

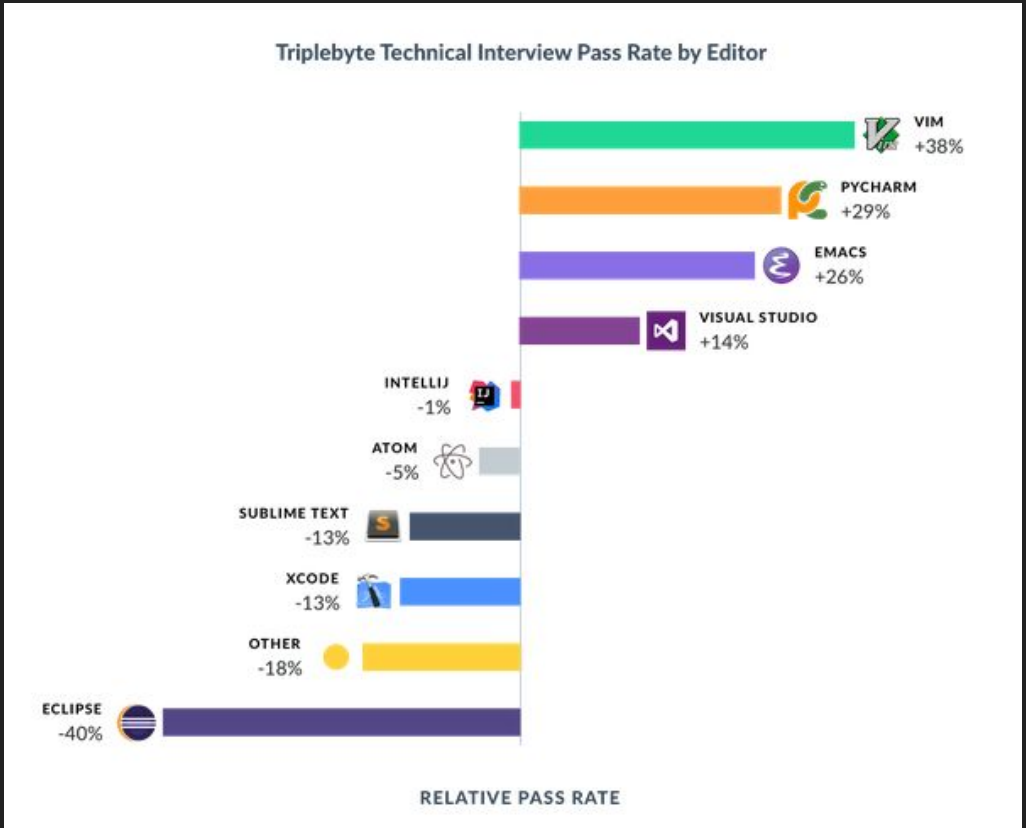


Editors


1 [No Name]
1
VIM - Vi IMproved
version 8.0.550
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable
Help poor children in Uganda!
type :help iccf<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info
NORMAL [No Name] 100% 0:1

Common Editors

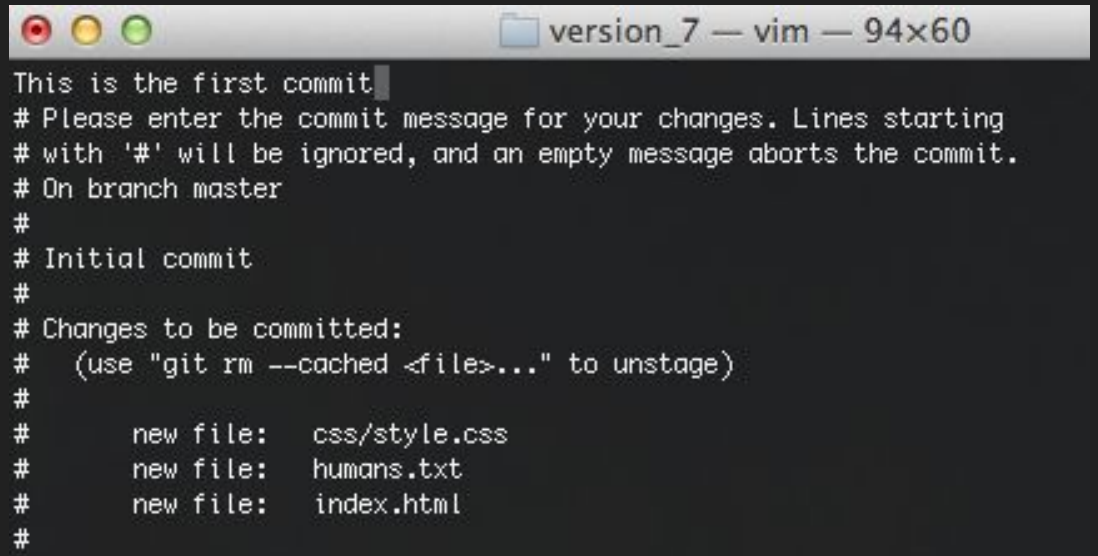
- Vim
- Emacs
- Nano (easy to use)



https://triplebyte.com/blog/technical-interview-performance-by-editor-os-language

Why vim?

- Available everywhere
- Productivity!



Basics of vim

- Moving around
- Yank and paste
- Panes

Tmux (Terminal Multiplexer)

- One terminal to potentially many
- Add windows to your terminal
- Save your sessions
- Screensharing, but better

Demo